

JUL 23 2007

Appl. No. 10/782,917  
Amdt. Dated July 23, 2007  
Reply to Office action of March 21, 2007

**Amendments to the Claims:**

This listing of claims will replace, without prejudice, all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently amended) A method of generating a target application from a reference application, the reference application being a Java application adapted to execute on a reference mobile device, the target application being configured for a target mobile device, the method comprising:
  - a) unpacking the reference application into a plurality of class files;
  - b) transforming the reference application into the target application by a plug-in, wherein the plug-in is adapted to transform a plurality of different reference applications into a corresponding plurality of target applications for a predetermined combination of the reference mobile device and the target mobile device, and wherein the reference application is in bytecode during the transformation step.
2. (Cancelled)
3. (Currently amended) The method of claim 12, wherein the plug-in comprises an instruction file and at least one library, wherein the transformation step comprises the instruction file instructing a transformation engine to modify a portion of the reference application with a selected software code stored in the library, wherein the portion of the reference application is not supported by the target mobile device.
4. (Cancelled)
5. (Currently amended) The method of claim 34, wherein the transformation step comprises adding a new class file to the reference application.

6. (Currently amended) The method of claim 34, wherein the modifying step comprises at least one action selected from the group of: adding a new method, renaming an existing method, replacing a first object method call with a second object method call, replacing the first object method call with a static method call, renaming a constant pool entry, and inserting a new inner class to an existing class.
7. (Original) The method of claim 6, further comprising saving the target application to a computer readable medium.
8. (Original) The method of claim 7, further comprising repeating step (a) and step (b) to transform the plurality of different reference applications into the plurality of corresponding target applications.
9. (Original) The method of claim 1, further comprising selecting a predetermined plug-in from a plurality of the plug-ins, the predetermined plug-in corresponding to the predetermined combination of the reference mobile device and the target mobile device, each of the plurality of the plug-ins corresponding to a different combination of the reference mobile device and the target mobile device.
10. (Original) The method of claim 3, further comprising repackaging the target application into executable code.
11. (Currently amended) The method of claim 10, wherein the repackaging step further comprises obfuscating the target the class files of the target application.
12. (Original) The method of claim 10, wherein the repackaging step further comprises pre-verifying the class files of the target application.
13. (Original) The method of claim 3, wherein the target application is a non-Java application.

14. (Original) The method of claim 6, wherein the unpacking step comprises unpacking a JAR file of the reference application.
15. (Original) The method of claim 14, wherein the method further comprises breaking an immutable image from the reference application into a plurality of smaller images.
16. (Currently amended) The method of claim 12, wherein the transformation step comprises inserting an interception module into the reference application, wherein the interception module is adapted to intercept key events.
17. (Original) The method of claim 16, wherein the transformation step comprises inserting a conversion table into the interception module, wherein the conversion table defines key-mapping from the reference mobile device to the target mobile device.
18. (Currently amended) A computer-implemented system for transforming a Java reference applications adapted to execute on a reference mobile device into corresponding target applications configured for a target mobile device, the system comprising:
- a) a transformation engine adapted to execute on a computer; and
  - b) a plug-in comprising:
    - i) an instruction file; and
    - ii) a selected software code adapted to modify a portion of the reference application not supported by the target mobile device;
- wherein the transformation engine is adapted to instruct the computer to access the instruction file, the instruction file being adapted to direct the transformation engine to identify the portion of the reference application and to instruct the computer to modify the portion with the selected software code to create the corresponding target application configured for the target mobile device, wherein the transformation engine is adapted to modify the reference application in bytecode.

19. (Original) The system of claim 18, wherein the instruction file comprises an XML file.
20. (Original) The system of claim 19, wherein the plug-in comprises a library, the library being adapted to store a plurality of the software codes adapted to modify a plurality of the portions.
21. (Original) The system of claim 20, wherein the transformation engine is a Java application.
22. (Original) The system of claim 19, wherein the selected software code comprises an interception module adapted to intercept key events.
23. (Original) The system of claim 22, wherein the interception module comprises a conversion table adapted to define key-mapping from the reference mobile device to the target mobile device.
24. (Original) The system of claim 19, wherein the selected software code comprises a new method adapted for insertion into the reference application.
25. (Original) The system of claim 19, wherein the selected software code comprises a static method call adapted for insertion into the reference application.
26. (Original) The system of claim 19, wherein the selected software code comprises a new method name adapted to replace a method name in the reference application.
27. (Original) The system of claim 19, wherein the selected software code comprises a new object method call adapted to replace a method call in the reference application.
28. (Original) The system of claim 19, wherein the selected software code comprises a new class file adapted for insertion into the reference application.

29. (Original) The system of claim 19, further comprising a plurality of the plug-ins, each of the plurality of the plug-ins corresponding to a different combination of reference and target mobile device, wherein the transformation engine is adapted to choose a selected one of the plug-ins corresponding to the different combination.

30. (Original) The system of claim 20, wherein the target applications are Java applications.

31. (Original) The system of claim 20, wherein the target applications are non-Java applications.